CLAIMS

Please amend the Claims as follows:

1. (Currently Amended) A method, for use in a computer system having a main memory and a first processor, ~~means~~ for providing said first processor with access to the register state of a supplemental processor, said supplemental processor otherwise inaccessible by said first processor, comprising:

loading a program into said supplemental processor;

executing said program in said supplementary processor to generate said register states of said supplementary processor;

storing said register states in said main memory; [[and]]

accessing said register state by said first processor; and

subsequent to storing said register states and accessing said register states, inspecting said register states for errors.

2. (Original) The method of claim 1 further including a program debugger operating on said main processor and wherein said method further includes:

accessing said register state in said main memory by said debugger to debug said program.

3. (Canceled).

4. (Currently Amended) [[The]] A method of debugging a specified program intended to operate on a supplemental processor, ~~said supplemental processor having limited memory and flexibility,~~ the register states of which cannot be directly accessed by a read command from another processor comprising:

reserving a pool of memory accessible to a debugging program processor;

running a specified program, to be debugged, in said supplemental processor until instructions in said specified program cause operation of said specified program to cease;

activating a secondary program in said supplemental processor to transmit register states of said supplemental processor, at the time said specified program is operationally interrupted, to said debugging program processor;

subsequent to transmitting said register states, inspecting said register states for errors;

modifying parameters of said specified program in a memory pool accessible by said debugging program processor; and

restoring operation of said specified program in said supplemental processor with ~~program alterations~~ modified parameters~~in the debugging process~~.

5. – 7. (Cancelled).

8. (Currently Amended) [[The]] A method of debugging a first processor unit employing a second processor operating a debugging program, comprising:

activating a secondary program in first processor to transmit register states of said first processor, said register states embodying program parameters, subsequent to a time a specified program to be debugged is operationally interrupted, to said second processor;

subsequent to transmitting said register states, inspecting said register states for errors;

modifying parameters of said specified program in a memory pool accessible by said second processor through the use of said debugging program in said second processor; and

restoring operation of said specified program in said first processor with alterations as created in the debugging process.

9. (Currently Amended) A computer program product for debugging a first processor employing a supplemental processor, the computer program product having a medium with a debugging computer program embodied thereon, the debugging computer program comprising:

computer code for transmitting and activating a secondary program in a first processor transmitting register states of said first processor, said register states embodying program parameters, subsequent to the operational halting of said program requiring debugging, to said second processor;

computer code for, subsequent to transmitting said register states, inspecting said register states for errors;

computer code for modifying parameters of said specified program in a memory pool accessible by said second processor through the use of said debugging program in said second processor; and

computer code for restoring operation of said specified program in said first processor with alterations generated by the debugging process.

10. (Currently Amended) A computer program product for authenticating code in a computer system, the computer program product having a medium with a computer program embodied thereon, the computer program comprising:

computer code for transmitting and activating a secondary program in a first processor transmitting register states of said first processor, said register states embodying program parameters, subsequent to the operational halting of said program requiring debugging, to said second processor;

computer code for, subsequent to transmitting said register states, inspecting said register states for errors;

computer code for modifying parameters of said specified program in a memory pool accessible by said second processor through the use of said debugging program in said second processor; and

computer code for restoring operation of said specified program in said first processor with alterations generated by the debugging process.

11. (New) A method for debugging a first computer program, the first computer program executable on a first processor in a computer system, wherein the first processor is a specialized function processing unit (SPU), the method comprising:

activating a debugging program on a main processing unit (MPU) of the computer system, the MPU inoperable to access registers of the SPU directly;

allocating, by the debugging program, a portion of a main memory of the computer system to store received SPU register contents;

verifying, by the debugging program, that the first computer program is halted;

activating, on the SPU, a second computer program;

transferring, by the second computer program, SPU register contents to the allocated portion of the main memory;

accessing, by the debugging program, the SPU register contents from the main memory;

subsequent to accessing the SPU register contents, inspecting the SPU register contents for errors;

modifying, by the debugging program, the SPU register contents in the main memory, in response to inspecting the SPU register contents;

transferring, by the second computer program, SPU register contents from the main memory to registers of the SPU; and

restarting, by the debugging program, the SPU.

12. (New) The method of Claim 12, further comprising transferring, by the second computer program, SPU state information to the MPU.

13. (New) The method of Claim 12, further comprising:

allocating, by the second computer program, a portion of a local store of the SPU;

copying, by the second computer program, SPU register contents to the allocated portion of the SPU local store; and

wherein transferring, by the second computer program, SPU register contents to the allocated portion of the main memory, comprises copying the SPU register contents from the allocated portion of the SPU local store to the allocated portion of the main memory.

14. (New) The method of Claim 13, wherein transferring, by the second computer program, SPU register contents from the main memory to registers of the SPU, further comprises:

copying the SPU register contents from the allocated portion of the main memory to the allocated portion of the SPU local store; and

loading the SPU register contents from the allocated portion of the SPU local store to the SPU registers.

15. (New) A computer program product for debugging a first computer program, the first computer program executable on a first processor in a computer system, wherein the first processor is a specialized function processing unit (SPU), the computer program product having a tangible computer-readable medium with a debugging computer program embodied thereon, the debugging computer program comprising:

computer code for activating a debugging program on a main processing unit (MPU) of the computer system, the MPU inoperable to access registers of the SPU directly;

computer code for allocating, by the debugging program, a portion of a main memory of the computer system to store received SPU register contents;

computer code for verifying, by the debugging program, that the first computer program is halted;

computer code for activating, on the SPU, a second computer program;

computer code for transferring, by the second computer program, SPU register contents to the allocated portion of the main memory;

computer code for accessing, by the debugging program, the SPU register contents from the main memory;

computer code for subsequent to accessing the SPU register contents, inspecting the SPU register contents for errors;

computer code for modifying, by the debugging program, the SPU register contents in the main memory, in response to inspecting the SPU register contents;

computer code for transferring, by the second computer program, SPU register contents from the main memory to registers of the SPU; and

computer code for restarting, by the debugging program, the SPU.

16. (New) The computer program product of Claim 15, further comprising:

computer code for allocating, by the second computer program, a portion of a local store of the SPU;

computer code for copying, by the second computer program, SPU register contents to the allocated portion of the SPU local store; and

wherein transferring, by the second computer program, SPU register contents to the allocated portion of the main memory, comprises copying the SPU register contents from the allocated portion of the SPU local store to the allocated portion of the main memory.

17. (New) The computer program product of Claim 16, wherein transferring, by the second computer program, SPU register contents from the main memory to registers of the SPU, further comprises:

copying the SPU register contents from the allocated portion of the main memory to the allocated portion of the SPU local store; and

loading the SPU register contents from the allocated portion of the SPU local store to the SPU registers.

18. (New) A debugging system, comprising:

a debugging program operational on a main processing unit (MPU) of a computer system, wherein the debugging program is a computer program;

the computer system comprising a main memory coupled to the MPU and a specialized function processing unit (SPU) coupled to the MPU and the main memory, wherein the MPU is inoperable to access registers of the SPU directly;

the SPU comprising a plurality of SPU registers and an SPU local store;

a target program operational on the SPU, wherein the target program is a computer program;

a copy program operational on the SPU, wherein the target program is a computer program;

wherein the copy program on the SPU is configured to:

transfer SPU register contents to an allocated portion of the main memory;

transfer SPU register contents from the main memory to registers of the SPU; and

wherein the debugging program on the MPU is further configured to:

allocate a portion of a main memory of the computer system to store received SPU register contents;

verify that the target computer program is halted;

activate, on the SPU, the copy program;

access the SPU register contents from the allocated portion of the main memory;

subsequent to accessing the SPU register contents, to inspect the SPU register contents for errors;

modify the SPU register contents in the main memory, in response to inspecting the SPU register contents; and

restart the SPU.

19. (New) The system of Claim 18, further comprising:

wherein the copy program is further configured to:

allocate a portion of a local store of the SPU;

copy SPU register contents to the allocated portion of the SPU local store; and

wherein transferring SPU register contents to the allocated portion of the main memory, comprises copying the SPU register contents from the allocated portion of the SPU local store to the allocated portion of the main memory.

20. (New) The system of Claim 19, further comprising:

wherein transferring SPU register contents from the main memory to registers of the SPU, further comprises:

copying the SPU register contents from the allocated portion of the main memory to the allocated portion of the SPU local store; and

loading the SPU register contents from the allocated portion of the SPU local store to the SPU registers.